



# Ciência de Dados I



## *Aula 03* – Leitura e Manipulação de Dados

---

Profa. Dra. Flávia Cristina M. Queiroz Mariano  
ICT – Unifesp, Campus São José dos Campos

[flavia.queiroz@unifesp.br](mailto:flavia.queiroz@unifesp.br)

Profa. Dra. Camila Bertini Martins  
EPM – Unifesp, Campus São Paulo

[cb.martins@unifesp.br](mailto:cb.martins@unifesp.br)



# OBJETIVOS DA AULA

- Ler um conjunto de dados (tabulados ou direto do Excel)
- Manipular um conjunto de dados para as análises
- Criar indicadores diretamente na base de dados.



# PROBLEMA

Quais as 6 (seis) cidades brasileiras com maior taxa de mortalidade por Covid-19?

$$\textit{Taxa de Mortalidade} = \frac{\text{n}^{\circ} \text{ de } \acute{o}\text{bitos}}{\text{n}^{\circ} \text{ de habitantes}} * 100.000$$

# PARA RESOLVER:

1. Juntar os bancos de dados disponíveis, IBGE\_pop com CovidBR;
2. Selecionar as variáveis Data, ibgeID, Cidade, Estado, Mortes, Casos, POP\_ESTIMADA;
3. Criar indicador de Mortalidade por cidade brasileira;
4. Ordenar (decrescentemente) o banco de dados pela taxa de mortalidade;
5. Filtrar o conjunto de dados pelas 6 primeiras observações.

# SOBRE O TIDYVERSE...

O pacote Tidyverse, na realidade, contém uma coleção de pacotes do R.



O intuito é proporcionar ao programador uma maior agilidade e produtividade.



Fonte: <https://www.tidyverse.org/>

# Instalação e Ativação do pacote

# Instalando o pacote

```
install.packages("tidyverse")
```

# Carregando/Ativando o pacote

```
library(tidyverse)
```

```
#require(tidyverse)
```

#Verificando os pacotes contidos no tidyverse

```
tidyverse_packages()
```

```
[1] "broom"      "cli"      "crayon"    "dbplyr"    "dplyr"
[6] "forcats"    "ggplot2"   "haven"     "hms"       "httr"
[11] "jsonlite"   "lubridate" "magrittr"   "modelr"    "pillar"
[16] "purrr"      "readr"     "readxl"    "reprex"    "rlang"
[21] "rstudioapi" "rvest"     "stringr"   "tibble"    "tidyr"
[26] "xml2"       "tidyverse"
```

# Ativando diretamente o dplyr

```
library(dplyr)
```



# Manipulação do conjunto de dados



## O que é o dplyr?

Documentação: <https://dplyr.tidyverse.org/> .

- É um poderoso pacote R para transformar e resumir dados tabulares com linhas e colunas. É considerado **o pacote mais importante** do tidyverse.

## Qual sua utilidade?

- Contém funções que permite a execução de operações comuns de manipulação de dados, como filtrar linhas, selecionar colunas específicas, reordenar linhas, adicionar novas colunas e resumir dados. Além de, dispor de uma função útil para concatenar mais de um tibble/data.frame.

## Qual vantagem em seu uso?

- Comparadas às funções básicas em R como `split()`, `subset()`, `apply()`, `sapply()`, `lapply()`, `tapply()` and `aggregate()`, as funções no dplyr são mais fáceis de trabalhar, mais consistentes na sintaxe e melhor direcionadas para análise de dados em cima de um `data.frame`, ao invés de trabalhar apenas com vetores.

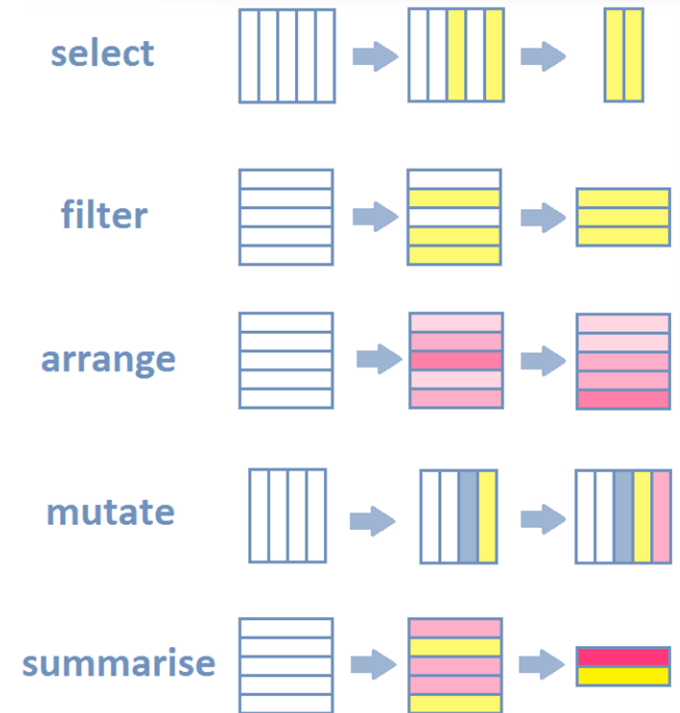


# Manipulação do conjunto de dados



Tabela. Principais funcionalidades do pacote dplyr.

Principais funções	Operação na manipulação de dados
<code>glimpse()</code>	Inspecionar o data.frame/tibble
<code>select()</code>	Selecionar colunas/variáveis
<code>filter()</code>	Filtrar linhas/observações
<code>arrange()</code>	Reordenar ou ordenar linhas da base
<code>mutate()</code>	Criar/modificar colunas
<code>group_by()</code>	Agrupar
<code>summarise()</code>	Sumarizar/Resumir (por um grupo específico)
<code>left_join()</code> , <code>right_join()</code> e <code>full_join()</code>	União de <i>tibbles</i> por coluna em comum





# Importando os dados



- Importação/leitura de dados tabulares (.csv, .tsv, .fwf)

Documentação: <https://readr.tidyverse.org/>.

```
a,b,c  
1,2,3  
4,5,NA
```

- `read_csv("file.csv")`    colunas delimitadas por vírgulas

```
a;b;c  
1;2;3  
4;5;NA
```

- `read_csv2("file2.csv")` colunas delimitadas por ponto e vírgula



- Importação/leitura de dados de arquivos do Excel

Documentação: <https://readr.tidyverse.org/>.

- `read_excel("file3.xls")`

- `read_excel("file4.xlsx")`



Mesma função  
para arquivos  
**.xls ou .xlsx**

# Importando os bancos de dados

---

# Arquivo .csv:

```
library(readr)
```

```
covid <- read_csv("cases-brazil-cities-time.csv", na = c("", "-", "NA"))
```

#ou do pacote base do R

```
covid2 <- read.csv("cases-brazil-cities-time.csv", stringsAsFactors = F, sep=",")
```

# Arquivo .xls ou .xlsx:

```
library(readxl)
```

```
pop.est <- read_excel("estimativa_TCU_2019_cidades.xlsx", sheet = 2)
```

# Características das variáveis



**glimpse**(covid)

```
Rows: 719,209
Columns: 12
$ date           <chr> "25/02/2020", "25/02/2020", "2...
$ country        <chr> "Brazil", "Brazil", "Brazil", ...
$ state          <chr> "SP", "TOTAL", "SP", "TOTAL", ...
$ city           <chr> "São Paulo/SP", "TOTAL", "São ...
$ ibgeID          <dbl> 3550308, 0, 3550308, 0, 355030...
$ newDeaths       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ deaths         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ newCases        <dbl> 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, ...
$ totalCases      <dbl> 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, ...
$ deaths_per_100k_inhabitants <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ totalCases_per_100k_inhabitants <dbl> 0.00816, 0.00048, 0.00816, 0.0...
$ deaths_by_totalCases <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
```

**glimpse**(pop.est)

```
Rows: 5,570
Columns: 5
$ UF             <chr> "RO", "RO", "RO", "RO", "RO", "RO", "RO", "RO", "RO", "RO", "RO", "RO", "R...
$ COD_UF         <dbl> 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11...
$ COD_MUNIC      <chr> "00015", "00023", "00031", "00049", "00056", "00064", "00072", "00080", "0...
$ MUNICIPIO      <chr> "Alta Floresta D'Oeste", "Ariquemes", "Cabixi", "Cacoal", "Cerejeiras", "C...
$ POP_ESTIMADA   <dbl> 22945, 107863, 5312, 85359, 16323, 15882, 7391, 18331, 32374, 46174, 51775...
```

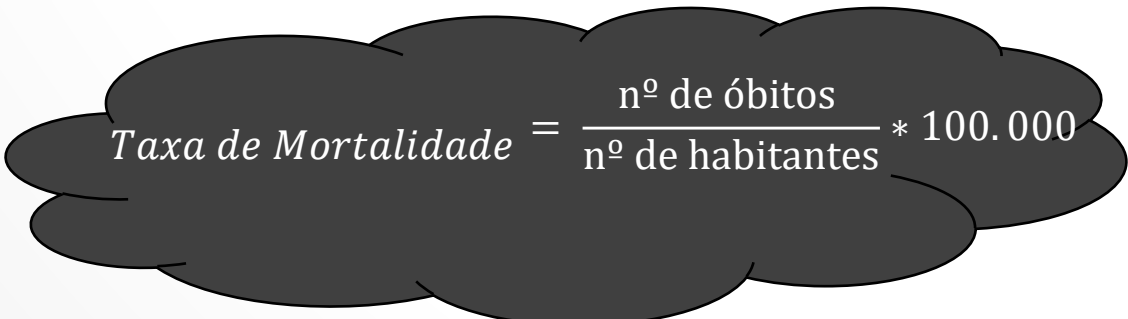
# Variáveis dos dois bancos de dados

## Covid -19

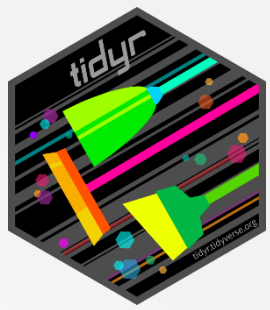
1. Data
2. País
3. Estado
4. Cidade
5. ID-IBGE
6. Novas mortes
7. Mortes
8. Novos casos
9. Total de Casos
10. Mortes por 100k habitantes
11. Total de Casos por 100k habitantes
12. Mortes por total de casos

## Pop. Estimada

1. UF
2. COD UF
3. COD\_MUNIC
4. MUNICIPIO
5. POP\_ESTIMADA


$$Taxa\ de\ Mortalidade = \frac{n^{\circ}\ de\ \acute{o}bitos}{n^{\circ}\ de\ habitantes} * 100.000$$

# União de duas colunas em uma só



```
library(tidyr)
```

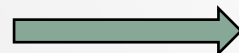
```
pop.est2<-unite(pop.est,ibgeID,COD_UF,COD_MUNIC, sep="")  
glimpse(pop.est2)
```

sem underscore(\_), que é o default,  
entre os valores de colunas diferentes  
→ Sem especificar esse argumento,  
teríamos, por exemplo, **35\_50308**

```
Rows: 5,570  
Columns: 4  
$ UF          <chr> "RO", "RO", "RO", "RO", "RO", "RO", "RO", "RO", "RO", "RO", "RO", "RO", "R...  
$ ibgeID      <chr> "1100015", "1100023", "1100031", "1100049", "1100056", "1100064", "1100072...  
$ MUNICIPIO   <chr> "Alta Floresta D'Oeste", "Ariquemes", "Cabixi", "Cacoal", "Cerejeiras", "C...  
$ POP_ESTIMADA <dbl> 22945, 107863, 5312, 85359, 16323, 15882, 7391, 18331, 32374, 46174, 51775...
```

MUNICIPIO	COD_UF	COD_MUNIC
São Paulo	35	50308
Rio de Janeiro	33	4557
Brasília	53	108
Salvador	29	27408
Fortaleza	23	4400
Belo Horizonte	31	6200

unite()



MUNICIPIO	ibgeID
São Paulo	3550308
Rio de Janeiro	334557
Brasília	53108
Salvador	2927408
Fortaleza	234400
Belo Horizonte	316200

COD\_UF  
COD\_MUNIC

chr

# Modificando/Criando uma variável



```
c2<-mutate(covid, ibgeID=factor(ibgeID))
```

mutate()

```
$ city  
$ ibgeID <dbl> "São Paulo/SP", "TOTAL", "São ...  
$ newDeaths <dbl> 3550308, 0, 3550308, 0, 355030...
```

glimpse(c2)

```
$ city  
$ ibgeID <fct> "São Paulo/SP", "TOTAL"  
$ newDeaths <int> 3550308, 0, 3550308, 0,
```

# Junção de conjuntos de dados

# Junção de Tibbles

```
dados<- inner_join(c2, pop.est2, by="ibgeID")
```



inner\_join()

state	city	ibgeID
SP	Sao Paulo/SP	3550308
RJ	Barra Mansa/RJ	3300407
BA	Feira de Santana/BA	2910800
RJ	Rio de Janeiro/RJ	3304557
SP	Sao Paulo/SP	3550308
DF	Brasilia/DF	5300108

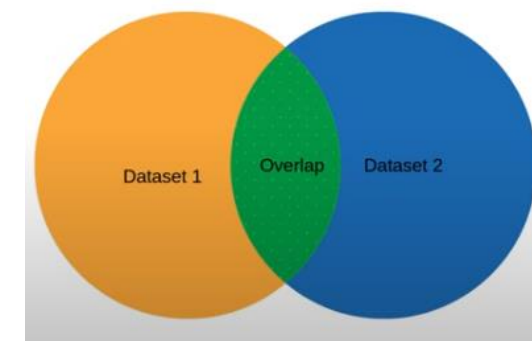
ibgeID	MUNICIPIO	POP_ESTIMADA
3550308	São Paulo	12.252.023
3304557	Rio de Janeiro	6.718.903
5300108	Brasília	3.015.268
3106200	Belo Horizonte	2.512.070
2910800	Feira de Santana	614.872



ibgeID foi a “chave” da junção

state	city	ibgeID	MUNICIPIO	POP_ESTIMADA
SP	Sao Paulo/SP	3550308	São Paulo	12.252.023
BA	Feira de Santana/BA	2910800	Feira de Santana	614.872
RJ	Rio de Janeiro/RJ	3304557	Rio de Janeiro	6.718.903
SP	Sao Paulo/SP	3550308	São Paulo	12.252.023
DF	Brasilia/DF	5300108	Brasília	3.015.268

## Outros tipos de joins:



Inner join: GREEN

Left join: YELLOW + GREEN

Right join: BLUE + GREEN

Full join: YELLOW + GREEN + BLUE

Semi join: GREEN - BLUE

Anti join: YELLOW - GREEN

Fonte: <https://www.youtube.com/watch?v=2W5-WrBEnEA>



# PIPE: operador %>%



Sem pipe:

```
verb(subject, complement)
```

Com pipe:

```
subject %>% verb(complement)
```



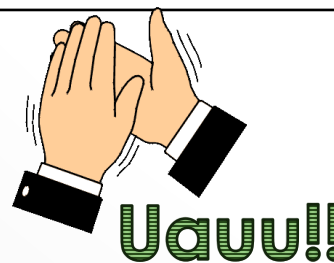
As funções disponíveis do tidyverse foram desenvolvidas para trabalhar com %>%, em que o conjunto de dados (neste caso, o subject) é o primeiro argumento dentro do comando após o %>%. **Este comando pode ser usado para *linkar* uma sequência de funções.**

```
dados<- inner_join(c2, pop.est2, by="ibgeID")
```

```
#library(magrittr)
```

```
dados<- c2 %>%
```

```
  inner_join(pop.est2, by="ibgeID")
```



# Selecionando variáveis



```
dados2<-dados%>%
```

```
  select(date, ibgeID, MUNICIPIO, UF, newDeaths, newCases, 16)
```

select()

```
columns: 15  
$ date  
$ country  
$ state  
$ city  
$ ibgeID  
$ newDeaths  
$ deaths  
$ newCases  
$ totalCases  
$ deaths_per_100k_inhabitants  
$ totalCases_per_100k_inhabitants  
$ deaths_by_totalCases  
$ UF  
$ MUNICIPIO  
$ POP_ESTIMADA
```



```
columns: 7  
$ date  
$ ibgeID  
$ MUNICIPIO  
$ UF  
$ newDeaths  
$ newCases  
$ POP_ESTIMADA
```



# Renomeando colunas



```
dados3<-dados2%>%
```

```
  select(Data=date, ibgeID, MUNICIPIO, UF, Mortes=newDeaths,  
         Casos=newCases, Pop=POP_ESTIMADA) #ou
```

```
select()
```

```
columns: 7  
$ date  
$ ibgeID  
$ MUNICIPIO  
$ UF  
$ newDeaths  
$ newCases  
$ POP_ESTIMADA
```

```
columns: 7  
$ Data  
$ ibgeID  
$ MUNICIPIO  
$ UF  
$ Mortes  
$ Casos  
$ Pop
```



```
#ou
```

```
colnames(dados2)<-c("Data","ibgeID", "MUNICIPIO", "UF", "Mortes", "Casos", "Pop")
```

# Sumarizando resultados por classe de uma variável



```
Total_city<-dados3 %>%  
  group_by(UF,MUNICIPIO)%>%  
  summarise(Mortes2=sum(Mortes),Pop=min(Pop))
```

group\_by()

summarise()

UF	MUNICIPIO	Mortes	Pop
RJ	Rio de Janeiro	1	6718903
SP	São Paulo	2	12252023
BA	Feira de Santana	1	614872
RJ	Rio de Janeiro	2	6718903
SP	São Paulo	5	12252023
DF	Brasília	0	3015268

UF	MUNICIPIO	Mortes	Pop
BA	Feira de Santana	1	614872
DF	Brasília	0	3015268
RJ	Rio de Janeiro	1	6718903
RJ	Rio de Janeiro	2	6718903
SP	São Paulo	5	12252023
SP	São Paulo	2	12252023

UF	MUNICIPIO	Mortes2	Pop
BA	Feira de Santana	1	614872
DF	Brasília	0	3015268
RJ	Rio de Janeiro	3	6718903
SP	São Paulo	7	12252023



# Modificando/Criando uma variável



```
TxMort<-Total_city %>%
```

```
  mutate(txMort=round((Mortes2*100000)/Pop,4))
```

mutate()

UF	MUNICIPIO	Mortes2	Pop
BA	Feira de Santana	1	614872
DF	Brasília	1	3015268
RJ	Rio de Janeiro	3	6718903
SP	São Paulo	7	12252023

UF	MUNICIPIO	Mortes2	Pop	txMort
BA	Feira de Santana	1	614872	0,1626
DF	Brasília	1	3015268	0,0332
RJ	Rio de Janeiro	3	6718903	0,0447
SP	São Paulo	7	12252023	0,0571

$$\text{Taxa de Mortalidade} = \frac{\text{n}^{\circ} \text{ de óbitos}}{\text{n}^{\circ} \text{ de habitantes}} * 100.000$$

# Filtrando por Classe/Grupo de variável



# Somente observações do **estado de SP** ;

TxMort %>%

```
filter(UF=="SP")
```

# Observações, **exceto da cidade de São Paulo**

TxMort %>%

```
filter(MUNICIPIO!="São Paulo")
```

filter()

UF	MUNICIPIO	Mortes2	Pop	txMort
BA	Feira de Santana	1	614872	0,1626
DF	Brasília	1	3015268	0,0332
RJ	Rio de Janeiro	3	6718903	0,0447
SP	São Paulo	7	12252023	0,0571
SP	Campinas	3	1204073	0,2492

UF	MUNICIPIO	Mortes2	Pop	txMort
SP	São Paulo	7	12252023	0,0571
SP	Campinas	3	1204073	0,2492

UF	MUNICIPIO	Mortes2	Pop	txMort
BA	Feira de Santana	1	614872	0,1626
DF	Brasília	1	3015268	0,0332
RJ	Rio de Janeiro	3	6718903	0,0447
SP	Campinas	3	1204073	0,0025

# (Re)ordenar linhas da base



#Ordem crescente

TxMort %>% **arrange**(UF, txMort)

#Ordem decrescente

TxMort %>% **arrange**(desc(txMort))

**arrange()**

UF	MUNICIPIO	Mortes2	Pop	txMort
BA	Feira de Santana	1	614872	0,1626
SP	São Paulo	7	12252023	0,0571
RJ	Rio de Janeiro	3	6718903	0,0447
RJ	Barra Mansa	1	184412	0,5423
SP	Campinas	3	1204073	0,2492

UF	MUNICIPIO	Mortes2	Pop	txMort
BA	Feira de Santana	1	614872	0,1626
RJ	Rio de Janeiro	3	6718903	0,0447
RJ	Barra Mansa	1	184412	0,5423
SP	São Paulo	7	12252023	0,0571
SP	Campinas	3	1204073	0,2492

UF	MUNICIPIO	Mortes2	Pop	txMort
RJ	Barra Mansa	1	184412	0,5423
SP	Campinas	3	1204073	0,2492
BA	Feira de Santana	1	614872	0,1626
SP	São Paulo	7	12252023	0,0571
RJ	Rio de Janeiro	3	6718903	0,0447



# PROBLEMA

Quais as 6 (seis) cidades brasileiras com maior taxa de mortalidade por Covid-19?

```
SeisMais <- TxMort %>%  
  arrange(desc(txMort))  
head(SeisMais)
```

**OBS.: A Tabela aqui apresentada trouxe a taxa de Mortalidade por 1000 (mil) habitantes.**

```
# A tibble: 6 x 5  
# Groups:   UF [5]  
   UF      MUNICIPIO      Mortes2      Pop      txMort  
  <chr>  <chr>          <dbl>    <dbl>    <dbl>  
1 RS      Charrua             11     3279     3.35  
2 RO      Pimenteiras do Oeste    6     2169     2.77  
3 SP      Gastão Vidigal          12     4808     2.50  
4 MT      São Pedro da Cipa        11     4727     2.33  
5 MT      Porto Esperidião        26    12017     2.16  
6 PI      Água Branca            36    17411     2.07
```

# PROBLEMA

Quais as 6 (seis) cidades brasileiras com maior taxa de mortalidade por Covid-19? Atualização dia 05 de Maio de 2021.

```
SeisMais<- TxMort %>%  
  arrange(desc(txMort))  
head(SeisMais)
```

**OBS.: A Tabela aqui apresentada trouxe a taxa de Mortalidade por 1000 (mil) habitantes.**

# A tibble: 6 x 5					
# Groups:   UF [5]					
	UF	MUNICIPIO	Mortes2	Pop	txMort
	<chr>	<chr>	<dbl>	<dbl>	<dbl>
1	RS	Charrua	11	3279	3.35
2	RO	Pimenteiras do Oeste	6	2169	2.77
3	SP	Gastão Vidigal	12	4808	2.50
4	MT	São Pedro da Cipa	11	4727	2.33
5	MT	Porto Esperidião	26	12017	2.16
6	PI	Água Branca	36	17411	2.07

# REFERÊNCIAS

- Principal referência utilizada neste arquivo:

- Wickham, Hadley; Grolemund, Garrett. **R for Data Science**: Import, Tidy, Transform, Visualize, and Model Data. 1st ed. O'Reilly Media. 2017.

- Documentação sobre pacotes:

- <https://www.tidyverse.org/>
- <https://readr.tidyverse.org/>
- <https://readxl.tidyverse.org/>
- <https://dplyr.tidyverse.org/>

- Conjunto de dados:

- <https://github.com/wcota/covid19br>
- <https://www.ibge.gov.br/estatisticas/sociais/populacao/9103-estimativas-de-populacao.html?=&t=downloads>

---

- Material complementar:

- <http://leg.ufpr.br/~walmes/cursoR/data-vis/slides/01-tidyverse.pdf>

